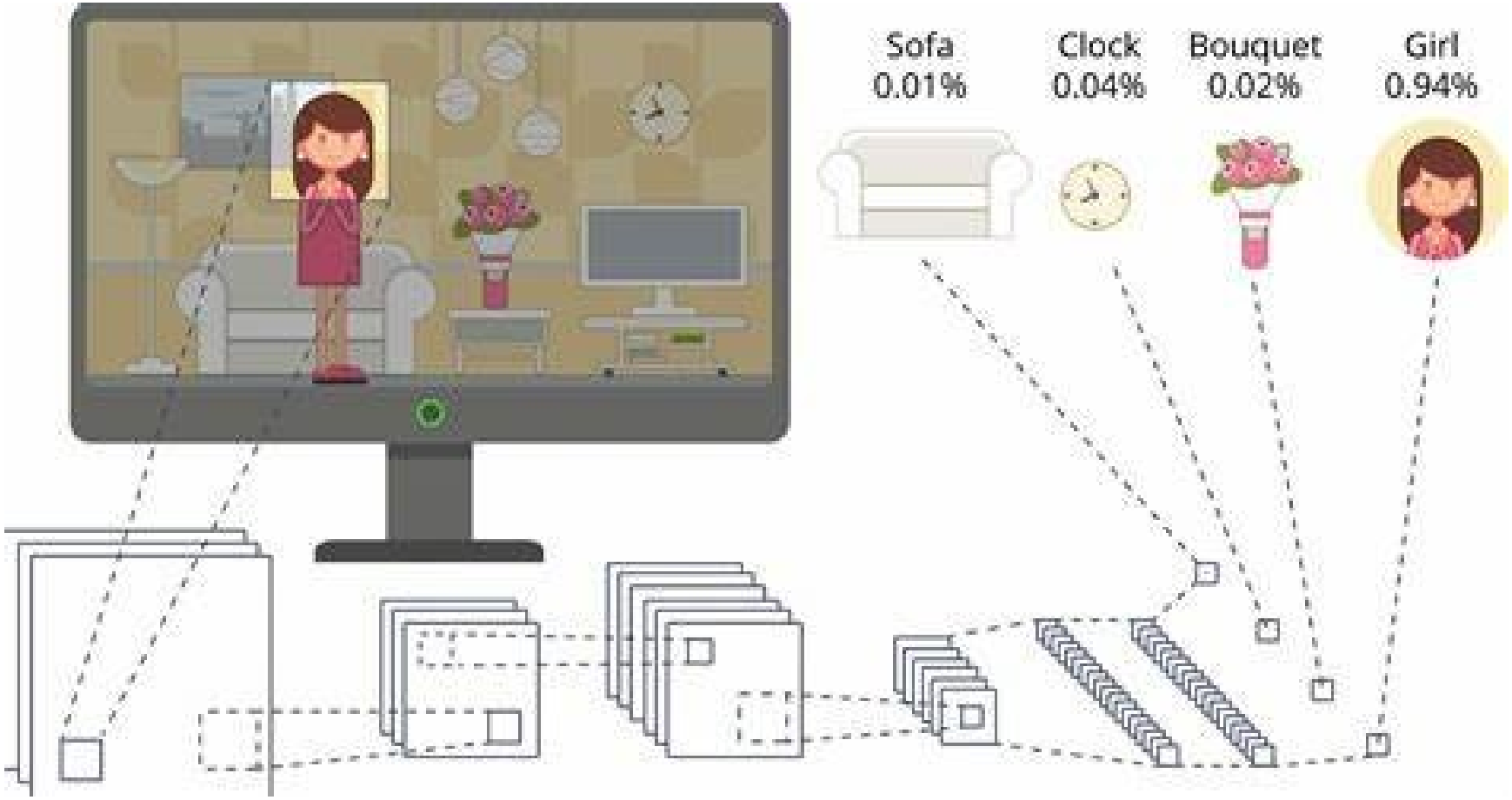
A7. Modelo convolucional.

XUNTA DE GALICIA
CONSELLERÍA DE CULTURA, EDUCACIÓN E UNIVERSIDADE

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
886 12 04 64
ies.teis@edu.xunta.es

GOBIERNO DE ESPAÑA · MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL
Unión Europea-NextGenerationEU



A7. Modelo convolucional.

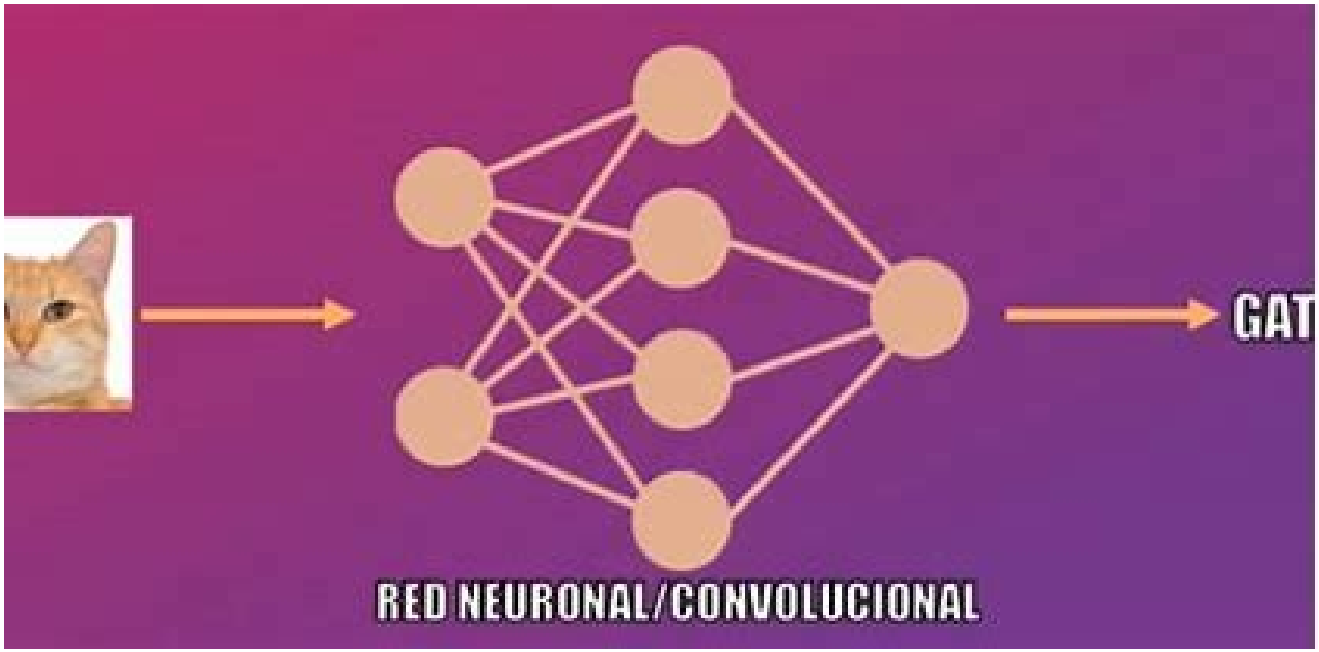
Índice.

A7. Modelo convolucional.

1. ¿Qué es una Red Neuronal Convolucional?

Las **Redes Neuronales Convolucionales** son un tipo de redes neuronales artificiales donde las "neuronas" corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico.

Este tipo de red es una variación de un **perceptrón multicapa**, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones.

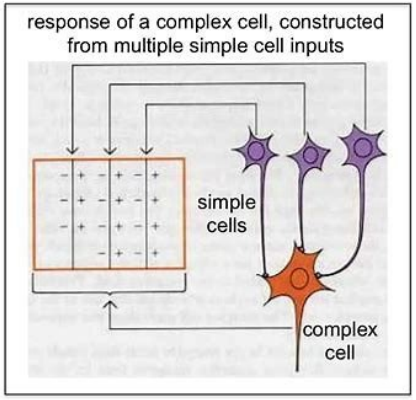


2. Semejanzas con el cerebro humano.

El trabajo realizado por Hubel y Wiesel en 1959 jugó ayudó en la comprensión del funcionamiento de la corteza visual, particularmente las células responsables de la selectividad de orientación y detección de bordes en los estímulos visuales dentro de la corteza visual primaria V1.

Dos tipos de células fueron identificadas debido a que tenían campos receptivos alargados, con lo cual tienen una mejor respuesta a los estímulos visuales alargados como las líneas y los bordes:

- Las **células simples** tienen regiones excitadoras e inhibitorias, que forman patrones elementales alargados en una dirección, posición y tamaño en particular en cada célula. Si un estímulo visual llega a la célula con la misma orientación y posición, de tal manera que ésta se alinea perfectamente con los patrones creados por las regiones excitadoras y al mismo tiempo se evita activar las regiones inhibitorias, la célula es activada y emite una señal.
- Las **células complejas** operan de una manera similar. Como las células simples, éstas tienen una orientación particular sobre la cual son sensibles. Sin embargo, éstas no tienen sensibilidad a la posición. Por ello, un estimulo visual necesita llegar únicamente en la orientación correcta para que esta célula sea activada.

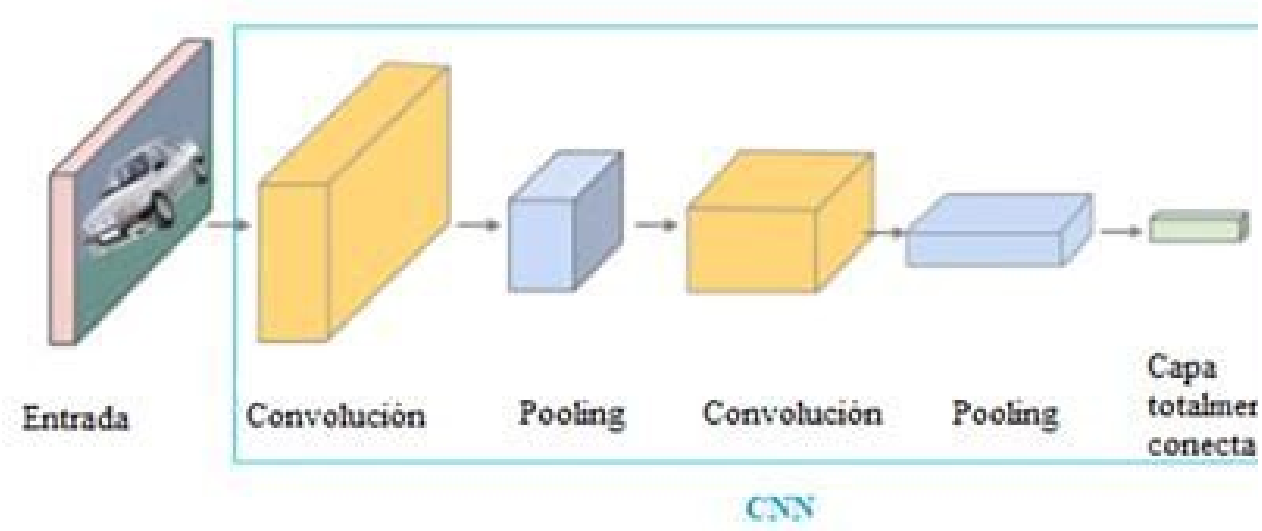


3. ¿Cómo están construidas y cómo funcionan?

Las **redes neuronales convolucionales** consisten en múltiples capas de filtros convolucionales de una o más dimensiones, a la que sigue, por lo general una función para realizar un mapeo causal no-lineal.

Como cualquier red empleada para clasificación, al principio estas redes tienen una fase de extracción de características, compuesta de **neuronas convolucionales**, luego hay una **reducción por muestreo** y al final tendremos **neuronas de perceptrón** mas sencillas para realizar la clasificación final sobre las características extraídas.

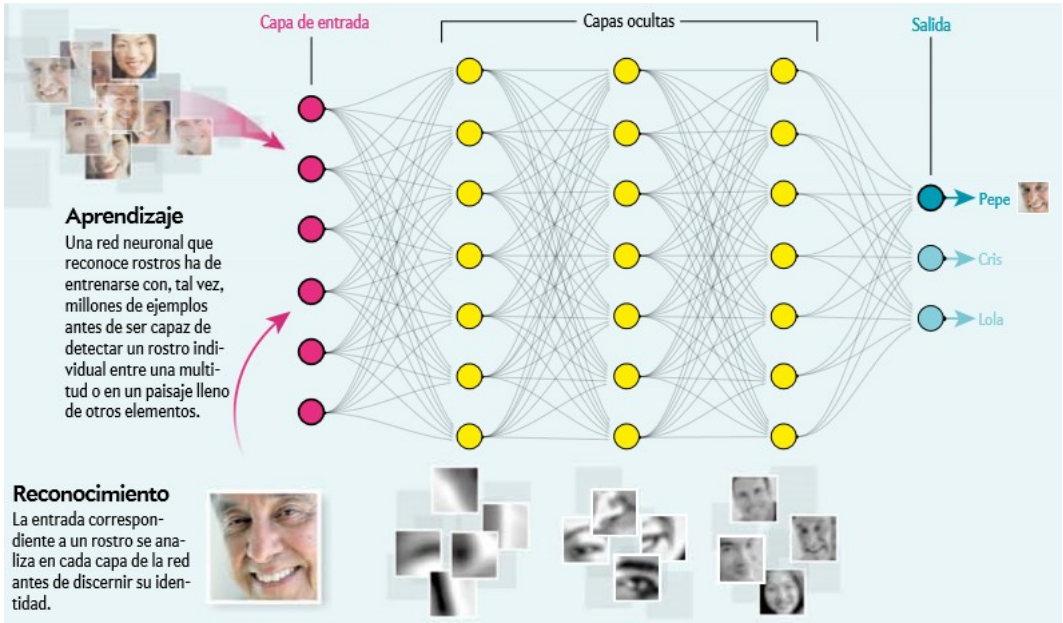
La **fase de extracción de características** se asemeja al proceso estimulante en las células de la corteza visual. Esta fase se compone de capas alternas de neuronas convolucionales y neuronas de reducción de muestreo. Según progresan los datos a lo largo de esta fase, se disminuye su dimensionalidad, siendo las neuronas en capas lejanas mucho menos sensibles a perturbaciones en los datos de entrada, pero al mismo tiempo siendo estas activadas por características cada vez más complejas.



4. ¿Cómo se logra que aprenda?

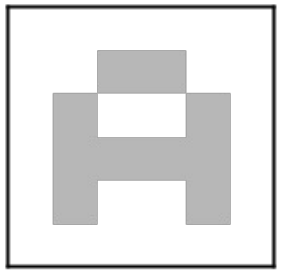
Las **Redes Neuronales Convolucionales** (CNN) aprenden a reconocer una diversidad de objetos dentro de imágenes, pero para ello necesitan "entrenarse" con una cantidad importante de "muestras" (más de 10.000), de ésta forma las neuronas de la red van a poder captar las características únicas de cada objeto, y a su vez, poder generalizarlo, a esto es lo que se le conoce como el **proceso de aprendizaje de un algoritmo**.

Nuestra red va a poder reconocer por ejemplo un cierto tipo de célula porque ya la ha "visto" anteriormente muchas veces, pero no solo buscará celulas semejantes sino que podrá inferir imagenes que no conozca pero que relaciona y en donde podrían existir similitudes , y esta es la parte inteligente del conocimiento

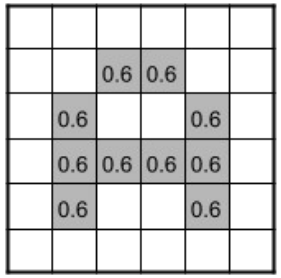


A7. Modelo convolucional.

4.1. Todo comienza con un pixel.



Una imagen...



...es una matriz de pixeles.
El valor de los pixeles va de 0 a 255 pero se normaliza para la red neuronal de 0 a 1

4.2. Píxeles y neuronas.

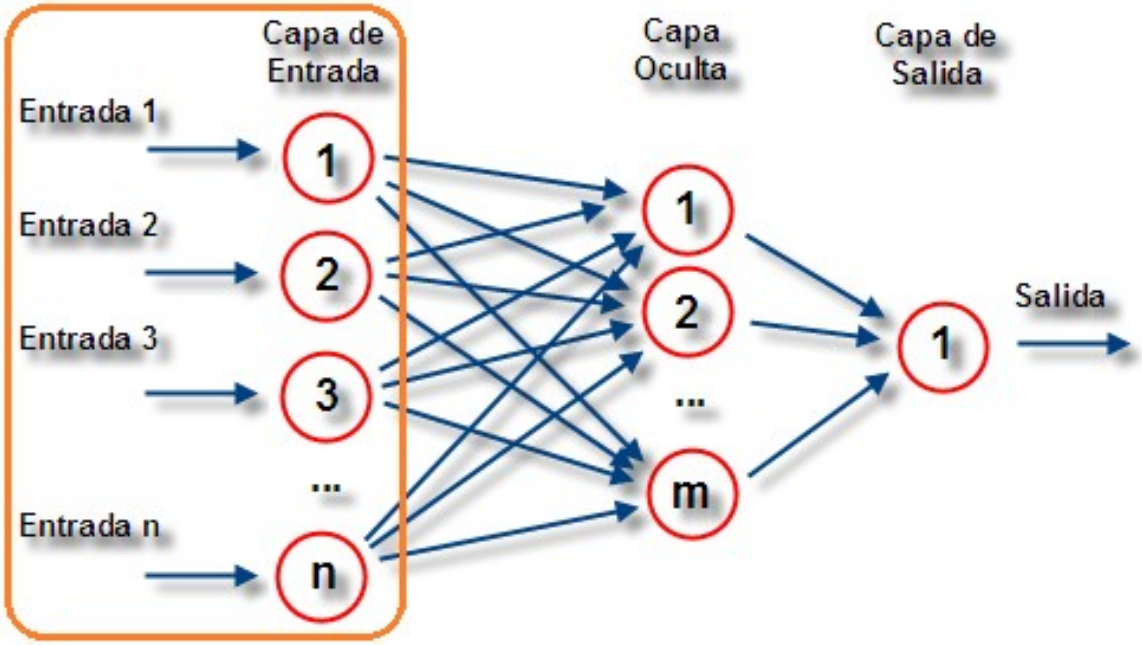
Para comenzar, la red toma como entrada los pixeles de una imagen.

Si tenemos una imagen con apenas 28×28 pixeles de alto y ancho, esto equivale a utilizar 784 neuronas.

Y eso es si sólo tenemos 1 color (escala de grises). Si tuviéramos una imagen a color, necesitaríamos 3 canales RGB (red, green, blue) y entonces usaríamos 28x28x3 = 2352 neuronas.

Estas neuronas constituyen nuestra capa de entrada.

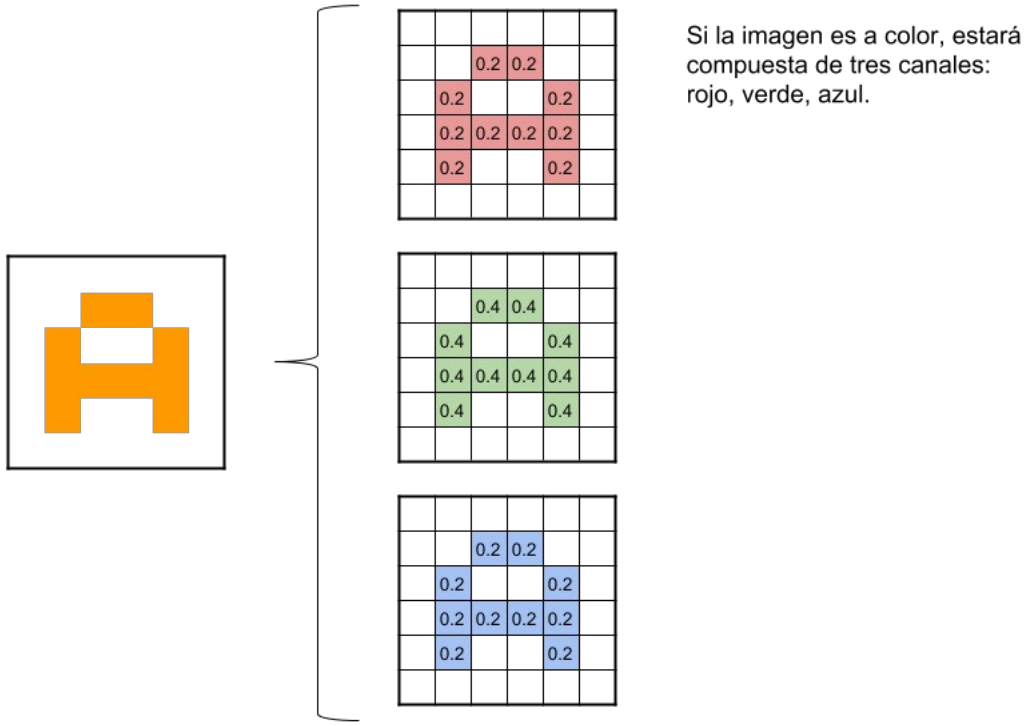
.



4.3. Pre-procesamiento.

Antes de alimentar la red, recuerda que como entrada nos conviene convertir los valores entre 0 y 1, y por tanto, tendremos que dividirlos todos entre 255.

Este 255 es por que los colores de los pixeles tienen valores que van del 0 al 255.



4.4. Convoluciones.

Ahora comienza el procesado distintivo de las **Redes neuronales Convolucionales**, es decir, las llamadas convoluciones: consistentes en tomar «grupos de pixeles cercanos» de la imagen de entrada e ir operando matemáticamente (producto escalar) contra una pequeña matriz que se llama kernel. Ese kernel supongamos que tiene un tamaño de de 3×3 pixels y con ese tamaño logra «visualizar» todas las neuronas de entrada (de izquierda-derecha, de arriba-abajo) y asi logra generar una nueva matriz de salida, que en definitiva será nuestra nueva capa de neuronas ocultas.

NOTA: si la imagen fuera a color, el kernel realmente sería de 3x3x3 es decir: un filtro con 3 kernels de 3×3; luego esos 3 filtros se suman (y se le suma una unidad bias) y conformarán 1 salida (cómo si fuera 1 solo canal).

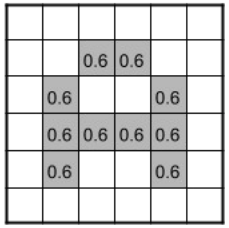


Imagen de entrada

1	0	-1
2	0	-2
1	0	-1

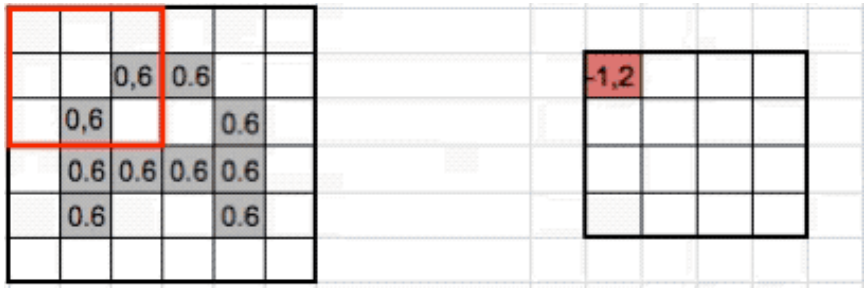
kernel

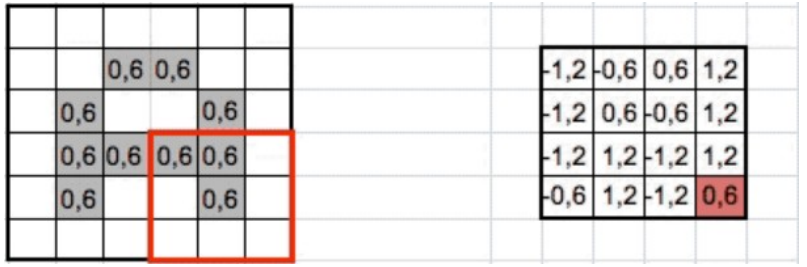
El kernel tomará inicialmente valores aleatorios y irá ajustando mediante backpropagation. Una mejora es hacer que siga una distribución normal siguiendo simetrías, pero sus valores son aleatorios.

4.5. Filtro: conjunto de kernels.

No aplicaremos 1 sólo kernel, si no que tendremos muchos kernel (al conjunto de Kernels se les llama filtros). Por ejemplo en esta primer convolución podríamos tener 32 filtros, con lo cual realmente obtendremos 32 matrices de salida (este conjunto se conoce como **feature mapping**), cada una de 28x28x1 dan un total del 25.088 neuronas para nuestra PRIMER CAPA OCULTA de neuronas, y que solo analiza una imagen cuadrada de apenas 28 pixeles?

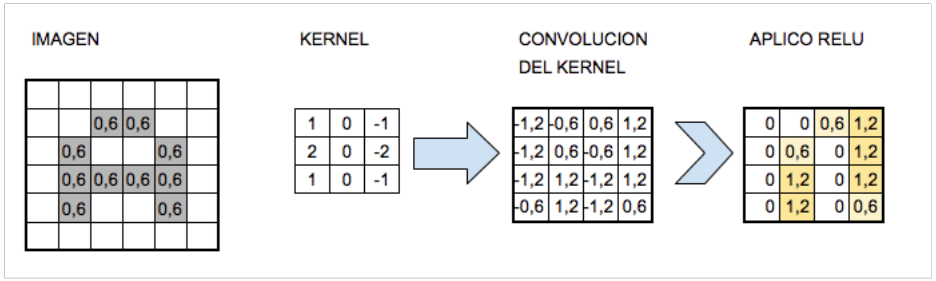
Imagina cuántas más serían si tomáramos una imagen de entrada de 224x224x3 (que aún es considerado un tamaño pequeño)…





4.5. Filtro: conjunto de kernels.

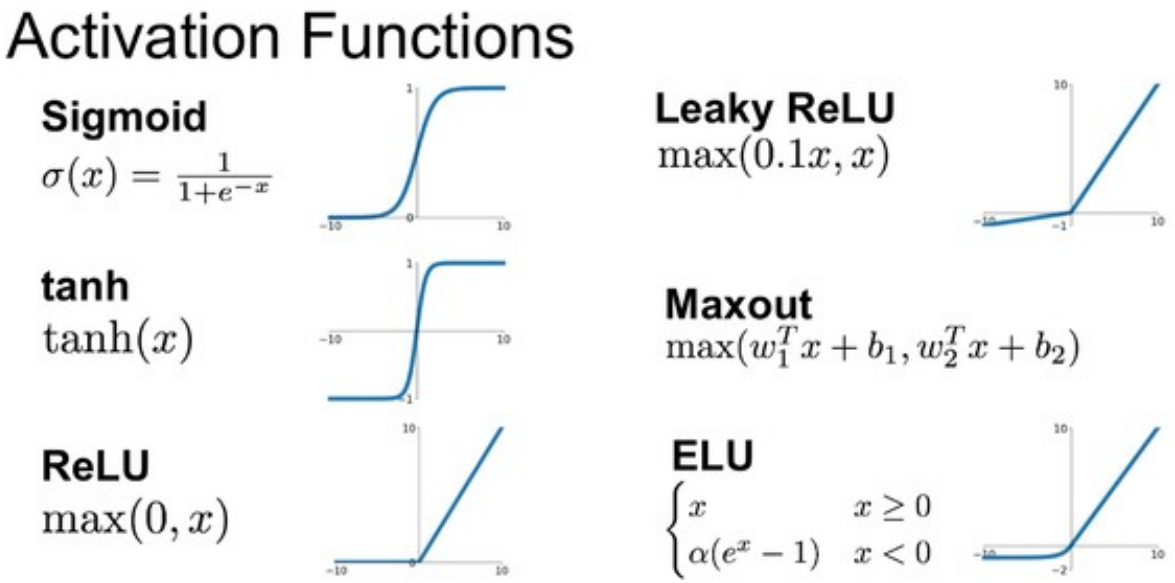
A medida que vamos desplazando el kernel y vamos obteniendo una **nueva imagen filtrada** por el kernel. En esta primer convolución y siguiendo con el ejemplo anterior, es como si obtuviéramos 32 imágenes filtradas nuevas. Estas imágenes nuevas lo que están dibujando son ciertas características de la imagen original. Esto ayudará en el futuro a poder distinguir un objeto de otro (por ej. gato ó un perro).



La imagen realiza una convolución con un kernel y aplica la función de activación, en este caso ReLu.

4.6. Función de Activación.

La función de activación más utilizada para este tipo de redes neuronales es la llamada ReLu por Rectifier Linear Unit y consiste en una función f(x)=max(0,x).



4.7. Muestreo (subsampling).

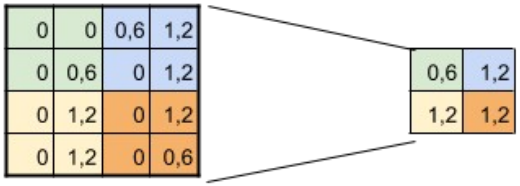
Ahora viene un paso en el que tomamos una muestra de las neuronas mas representativas antes de hacer una nueva convolución.

¿Por qué muestreamos ? Antes vimos que con imagen blanco y negro de 28x28pixels tenemos una primer capa de entrada de 784 neuronas y luego de la primer convolución obtenemos una capa oculta de 25.088 neuronas -que realmente son nuestros 32 mapas de características de 28×28 ((28 x 28) x 32)

Si hiciéramos una nueva convolución a partir de esta capa, el número de neuronas de la próxima capa requeriría un poder computacional importante. Por ello y para reducir el tamaño de la próxima capa de neuronas hacemos un muestreo preservando las características más importantes que detectó cada filtro.

Hay diversos tipos de muestreo (subsampling) , El «más usado» es: Max-Pooling

4.8. Muestreo con Max-Pooling.



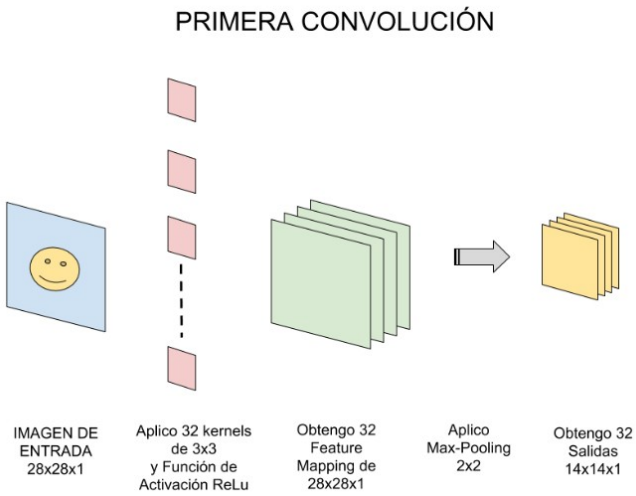
SUBSAMPLING:
Aplico Max-Pooling de 2x2
y reduzco mi salida a la mitad

El paso siguiente sería utilizar la tecnica de "Max-pooling" con un tamaño de 2×2.

Esto quiere decir que recorreremos cada una de las 32 imágenes de características obtenidas anteriormente de 28x28px de izquierda-derecha, arriba-abajo PERO en vez de tomar de a 1 pixel, tomaremos de «2×2» (2 de alto por 2 de ancho = 4 pixeles) e iremos preservando el valor «más alto» de entre esos 4 pixeles (por eso lo de «Max»). En este caso, usando 2×2, la imagen resultante es reducida «a la mitad»y quedará de 14×14 pixeles.

Luego de este proceso de submuestreo nos quedarán 32 imágenes de 14×14, pasando de 25.088 neuronas a 6272, las cuales son bastantes menos y que, en teoría, deberían seguir almacenando la información más importante para detectar características deseadas.

4.9. Convoluciones subsecuentes.



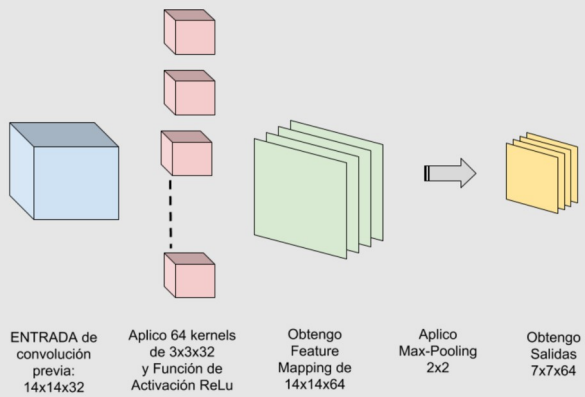
La imagen superior representa esa primera convolución: consiste de una entrada, un conjunto de filtros, generamos un mapa de características y hacemos el submuestreo. Con lo cual, en el ejemplo de imágenes de 1 sólo color tendremos:

PRIMERA CONVOLUCION	2)Aplico Kernel	3)Obtengo Feature Mapping	4)Aplico Max-Pooling	5)Obtengo «Salida» de la 1era Convolución
28x28x1 = 784 neuronas	32 filtros de 3×3	**28x28x 32 kernel = 25.088 neuronas**	de 2×2	**14x14x32 = 6.272 neuronas**

La primer convolución es capaz de detectar características primitivas como lineas ó curvas. A medida que hagamos más capas con las convoluciones, los mapas de características serán capaces de reconocer formas más complejas, y el conjunto total de capas de convoluciones podrá «reconocer».

4.9. Convoluciones subsecuentes.

Ahora deberemos hacer una Segunda convolución que será:



SEGUNDA CONVOLUCION	2)Aplico Kernel	3)Obtengo Feature Mapping	4)Aplico Max-Pooling	5)Obtengo «Salida» de la Convolución
14x14x32= 6272 neuronas	64 filtros de 3×3	**14x14x64 = 12544 neuronas**	de 2×2	**7x7x64 = 3136 neuronas**

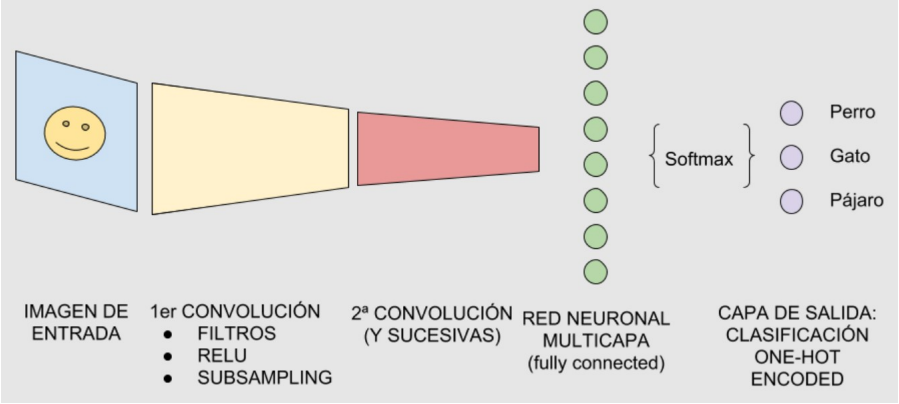
La 3ª convolución comenzará en tamaño 7×7 pixels y luego del max-pooling quedará en 3×3 con lo cual podríamos hacer sólo 1 convolución más. En este ejemplo empezamos con una imagen de 28x28px e hicimos 3 convoluciones. Si la imagen inicial hubiese sido mayor (de 224x224px) aún hubiéramos podido seguir haciendo convoluciones.

TERCERA CONVOLUCION	2)Aplico Kernel	3)Obtengo Feature Mapping	4)Aplico Max-Pooling	5)Obtengo «Salida» de la Convolución
7x7x64= 3.136 neuronas	128 filtros de 3×3	**7x7x128= 6272 neuronas**	de 2×2	**3x3x128 = 768 neuronas**

Y con esto ya hemos llegado a la última convolución.

4.10. Conexión con una red neuronal tradicional.

Para terminar, tomaremos la última capa oculta a la que hicimos subsampling, que se dice que es «tridimensional» por tomar la forma -en nuestro ejemplo- 3x3x128 (alto,ancho,mapas) y la «aplanamos», esto es que deja de ser tridimensional, y pasa a ser una capa de neuronas «tradicionales», y con ello aplanamos (y conectamos) una nueva capa oculta de neuronas tipo feedforward.



Entonces, a esta nueva capa oculta «tradicional», le aplicamos una función llamada Softmax que conecta contra la capa de salida final que tendrá la cantidad de neuronas correspondientes con las clases que estamos clasificando. Si perros y gatos, 2 neuronas, si coches, aviones ó barcos serán 3, etc.

Las salidas al momento del entrenamiento tendrán el formato conocido como «one-hot-encoding» en el que para perros y gatos sera: [1,0] y [0,1], para coches, aviones ó barcos será [1,0,0]; [0,1,0];[0,0,1].

Y la función de Softmax se encarga de pasar a probabilidad (entre 0 y 1) a las neuronas de salida. Por ejemplo una salida [0,2 0,8] nos indica 20% probabilidades de que sea perro y 80% de que sea gato, segun este ejemplo.

4.11. CNN y el modelo matemático de backpropagation de las redes neuronales.

El proceso es similar al de las redes tradicionales en las que tenemos una entrada y una salida esperada (por eso le llamamos aprendizaje supervisado) y mediante ese procesamiento de ajuste hacia adelante y hacias atras, vamos mejorando el valor de los pesos de las interconexiones entre capas de neuronas y a medida que iteramos esos pesos se ajustan hasta ser óptimos.

En el caso de la CNN, deberemos **ajustar el valor de los pesos de los distintos kernels**. Esto es una gran ventaja al momento del proceso de aprendizaje pues como vimos cada kernel es de un tamaño reducido, en nuestro ejemplo en la primer convolución es de tamaño de 3×3, eso son sólo 9 parámetros que debemos ajustar en 32 filtros dan un total de 288 parámetros. En comparación con los pesos entre dos capas de neuronas «tradicionales»: una de 748 y otra de 6272 en donde están TODAS interconectadas con TODAS y **eso equivaldría a tener que entrenar y ajustar más de 4,5 millones de pesos** (sólo para 1 capa).

A7. Modelo convolucional.

5. Comparativa con una red neuronal tradicional.

CARACTERISTISTICAS	Red «tradicional» Feedforward multicapa	Red Neuronal Convolucional CNN
Datos de entrada en la Capa Inicial	Las características que analizamos. Por ejemplo: ancho, alto, grosor, etc.	Pixeles de una imagen. Si es color, serán 3 capas para rojo,verde,azul
Capas ocultas	elegimos una cantidad de neuronas para las capas ocultas.	Tenemos de tipo: * Convolución (con un tamaño de kernel y una cantidad de filtros) * Subsampling
Capa de Salida	La cantidad de neuronas que queremos clasificar. Para «comprar» ó «alquilar» serán 2 neuronas.	Debemos «aplanar» la última convolución con una (ó más) capas de neuronas ocultas «tradicionales» y hacer una salida mediante SoftMax a la capa de salida que clasifica «perro» y «gato» serán 2 neuronas.
Aprendizaje	Supervisado	Supervisado
Interconexiones	Entre capas, todas las neuronas de una capa con la siguiente.	Son muchas menos conexiones necesarias, pues realmente los pesos que ajustamos serán los de los filtros/kernels que usamos.
Significado de la cantidad de capas ocultas	Realmente es algo desconocido y no representa algo en sí mismo.	Las capas ocultas son mapas de detección de características de la imagen y tienen jerarquía: primeras capas detectan lineas, luego curvas y formas cada vez más elaboradas.
Backpropagation	Se utiliza para ajustar los pesos de todas las interconexiones de las capas	Se utiliza para ajustar los pesos de los kernels.

6. Resumen de arquitectura básica.

La aqruitectura básica de un modelo Convolucional, se puede resumir en los siguientes componentes:

- **Entrada**: Serán los pixeles de la imagen. Serán alto, ancho y profundidad será 1 sólo color o 3 para Red,Green,Blue.
- **Capa de Convolución**: procesará la salida de neuronas que están conectadas en «regiones locales» de entrada (es decir pixeles cercanos), calculando el producto escalar entre sus pesos (valor de pixel) y una pequeña región a la que están conectados en el volumen de entrada. Aquí usaremos por ejemplo 32 filtros o la cantidad que decidamos y ese será el volumen de salida.
- **CAPA RELU** aplicará la función de activación en los elementos de la matriz.
- **MUESTREO ó SUBSAMPLING**: Hará una reducción en las dimensiones alto y ancho, pero se mantiene la profundidad.
- **CAPA TRADICIONAL** red de neuronas feedforward que conectará con la última capa de subsampling y finalizará con la cantidad de neuronas que queremos clasificar.